

# Studienarbeit Informationsvisualisierung 2021 – Rubenbauer Franziska

Auf der Suche nach Corona Datensätze habe ich mich nach kurzer Zeit für die RKI Webseite entschieden, da diese vertrauenswürdig ist und ihre Daten frei zur Verfügung stellt. Die Daten sind auf Deutschland bezogen. Besonders interessant fand ich hier die „COVID-19 Fälle nach Altersgruppe und Meldewoche“. Diese Daten sind aus einem Bereich, von dem man noch nicht viel gehört oder gesehen hat, was diese so interessant macht.

Quelle:

[https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Daten/Altersverteilung.html;jsessionid=EC085F189E46636FFABE01920AF9C225.internet072?nn=13490888](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Altersverteilung.html;jsessionid=EC085F189E46636FFABE01920AF9C225.internet072?nn=13490888)

## *Data Transformation:*

Der Datensatz steht als Excel Tabelle mit zwei Arbeitsmappen zur Verfügung. Diese habe ich händisch getrennt, weil das Python nicht kann. Zuerst habe ich mir die Altersinzidenz pro Kalenderwoche und Altersgruppe angeschaut. Einen ersten einfachen Überblick liefert das Balkendiagramm. Hierbei ist eins am Beginn des Datensatzes und eins am Ende angesiedelt. Dabei fällt schon auf, dass die Inzidenzwerte der jüngeren Bevölkerung übers Jahr stark gestiegen sind. Diese Vermutung habe ich mir dann genauer angeschaut und weitere Balkendiagramm zu verschiedenen Zeitpunkten gezeichnet. Dabei scheint sich die Vermutung zu bestätigen.

Danach warf ich einen Blick in den anderen Datensatz, welcher die genauen Fallzahlen nach Altersgruppen enthält. Nach einem unübersichtlichen Liniendiagramm entschied ich mir für den Datensatz mit der Altersinzidenz. Dabei wollte ich den Anteil, den die Altersgruppeninzidenz von der Ganzen einnimmt wissen. Dafür würde sich ein Kuchendiagramm anbieten, was nach einigen Versuchen auch recht gut aussieht.

Damit kam ich zu dem Entschluss, die Veränderung der Anteile der Altersgruppen an der Inzidenz darstellen zu wollen.

## *Visual Mapping:*

Der prozentuale Anteil der Altersgruppeninzidenz am Gesamten soll sich in der Breite des entsprechenden Kreissegments widerspiegeln. Dafür eignet sich das Kuchendiagramm, da man auf dem ersten Blick sieht, wie sich ein Teil im Vergleich zum Ganzen verhält. Bei einem Linien oder Balkendiagramm ist das nicht ganz so auffällig.

Die Altersgruppen sollen sich in der Farbe unterscheiden. Das ist das Naheliegendste, da die Farbe konstant einer Altersgruppe zugeordnet werden kann.

Die zeitliche Veränderung der Anteile soll in einer Animation dargestellt. Ein bestimmtes Zeitintervall sollen die Daten für einen Zeitpunkt dargestellt werden und im nächsten Intervall die Daten des nächsten Zeitpunktes. Dadurch wird die Veränderung der Anteile in Abhängigkeit von der Zeit visualisiert.

## *Implementierung:*

Angefangen habe ich mit der Fragestellung, wie ich die Daten aus der csv-Datei in meiner JavaScript verwenden kann. Dazu habe ich einen frei verfügbaren Parser aus dem Internet verwendet (Quelle: [www.papaparse.com](http://www.papaparse.com)). Dieser speichert alle Daten in ein großes Array. Daraus lese ich die Daten, welche für mich wichtig sind, in Variablen ein. Dazu verwende ich eine For-Schleife, die über die gesamte Länge des Arrays läuft, jedoch bei 2 anfängt, da die 0. und 1. Zeile dem Header und der Gesamtzeile besteht. Die Altersgruppen in der Tabelle verlaufen in 5er Schritten, was jedoch beim

```
function ladenVar(data){
  for(let j=2; j<data[0].length; j++){
    {
      zeit.push(data[0][j]);
      altersgruppe90.push(parseFloat(data[2][j]));
      //Zusammenfassen von z.B. 80-85 mit 85-89 zu AG80
      altersgruppe80.push((parseFloat(data[3][j]))+(parseFloat(data[4][j])));
      altersgruppe70.push((parseFloat(data[5][j]))+(parseFloat(data[6][j])));
      altersgruppe60.push((parseFloat(data[7][j]))+(parseFloat(data[8][j])));
      altersgruppe50.push((parseFloat(data[9][j]))+(parseFloat(data[10][j])));
      altersgruppe40.push((parseFloat(data[11][j]))+(parseFloat(data[12][j])));
      altersgruppe30.push((parseFloat(data[13][j]))+(parseFloat(data[14][j])));
      altersgruppe20.push((parseFloat(data[15][j]))+(parseFloat(data[16][j])));
      altersgruppe10.push((parseFloat(data[17][j]))+(parseFloat(data[18][j])));
      altersgruppe0.push((parseFloat(data[19][j]))+(parseFloat(data[20][j])));
    }
    for(let i=0; i<altersgruppe0.length;i++){
      //Gesamtinzidenz selbst ermitteln
      let tmp=altersgruppe0[i]+altersgruppe10[i]+altersgruppe20[i]+altersgruppe30[i]+
      altersgruppe40[i]+altersgruppe50[i]+altersgruppe60[i]+altersgruppe70[i]+altersgruppe80[i]+altersgruppe90[i];
      gesamt.push(tmp);
    }
  }
}
```

Diagramm zu einer zu großen Splitterung führen würde. Deswegen fasse ich die Altersgruppen in 10er Schritten zusammen. Damit die Daten aber dazu passen, addiert man diese z.B. von der Gruppe 60-65 mit denen der Gruppe 65-70. Um für das Diagramm später den Anteil berechnen zu können, brauchen die Gesamtsumme. Hierzu eignet sich die „Gesamt“-Zeile aus den Daten nicht, da diese nicht die Summe ist, sondern die gesamte Inzidenz pro 100.000

beschreibt. Den Unterschied erkläre ich mir so: Die Altersgruppeninzidenz verrechnet die Anzahl der positiven Fälle mit der Anzahl der Menschen in dieser Altersgruppe. Deswegen ist die Summe mehr als die Gesamtinzidenz in der Tabelle.

Zu besseren Lesbarkeit verändere ich anschließend das Format der Zeitangabe von z.B. „2020\_11“ (Tabelle) in „KW 11 2020“ mit entsprechenden String Funktionen. Damit ist die Verarbeitung der Daten soweit abgeschlossen.

Als nächsten Schritt habe ich mit einem einfachen statischen Kuchendiagramm begonnen. Dazu hat sich ein Beispiel auf der offiziellen Webseite von p5 angeboten. (Quelle: <https://p5js.org/examples/form-pie-chart.html> ).

Für die einzelnen Kreissegmente berechne ich mir zuerst den Anteil am Ganzen, um diesen dann als Breite verwenden zu können. Dies muss für jede Altersgruppe gemacht werden, wobei das Segment immer beim letzten Winkel nicht mehr bei 0 anfangen soll.

```
250 fill(112,112,112);
251 anteil=(ag0*100/gesamt); //Anteil am ganzen Kreis berechnen
252 arc(width/2, height/2, durchmesser, durchmesser, letzterWinkel, letzterWinkel+radians(anteil),PIE);
253 if(zusatzInfos == true){ //Zahlen am Rand zeichnen
254   push();
255   translate(width/2,height/2);
256   let x=(cos(radians(anteil)/2))*(durchmesser/2);
257   let y=(sin(radians(anteil)/2))*(durchmesser/2)+20;
258   let tmp=(ag0*100/gesamt);
259   tmp=round(tmp)+"%";
260   text(tmp, x, y);
261   pop();
262 }
```

Da das aber im Moment nur einen einzelnen Zeitpunkt gerecht wird, fehlt das Verhalten mit der Zeit noch. Dazu nutze ich aus, dass die draw()-Funktion bei jedem Frame neu ausgeführt wird. So kann ich bei jedem Durchlauf einen anderen Zeitpunkt nehmen. Dazu hab ich das Zeichnen der Kreissegmente in eine Funktion ausgelagert und übergebe mit jedem Durchlauf andere Array-Werte. Dazu dient eine Laufvariable „i“, die außerhalb von draw() initialisiert wird, da sie sonst jedes Mal zurückgesetzt werden würde. Am Ende von draw() wird sie jedes Mal um Eins erhöht und nachdem sie so groß ist wie die Länge des Datensatzes auf 0 zurückgesetzt um von vorne starten zu können. Damit ist die grundlegende Animation fertig des Kreisdiagramms fertig.

```
484 legendeY+=30;
485 fill(112,112,112);
486 rect(legendeX, legendeY, 15,15);
487 fill(0,0,0);
488 text("0 - 9", legendeX+20, legendeY+15);
489 if(zusatzInfos2 == true){
490   legendeY+=20;
491   let tmp = "Inzidenz: "+runden(ag0);
492   text(tmp,legendeX+15, legendeY+15);
493 }
```

Als nächste brauchte ich noch eine Legende, die verrät, welche Altersgruppe zu welcher Farbe gehört. Dazu zeichne ich ein Rechteck in der Farbe des Kreissektors und setze daneben den Text. Um das zu erleichtern, verschiebe ich den Nullpunkt des Koordinatensystems in die rechte obere Ecke der Zeichenfläche. Da man durch einen Buttonklick die

absoluten Zahlen anzeigen lassen können soll, habe ich auch diese Zeichnung in eine Funktion ausgelagert und verwende das gleiche System wie beim Diagramm.

Jetzt fehlt noch jede Indikation, welcher Zeitpunkt gerade dargestellt wird. Diesen will ich in der Mitte des Diagramms einfügen. Dazu zeichne ich zuerst einen kleineren Kreis in die Mitte des Diagramms, welcher die gleiche Farbe wie der Hintergrund besitzt. Anschließend gebe ich mir den entsprechenden Zeitpunkt darin aus. Auch hier sollen sich zusätzliche Informationen anzeigen lassen, welche durch eine if-Anweisung nur dann gezeichnet werden, wenn gewollt.

Da ich persönlich finde, dass man mit Kalenderwochen weniger anfangen kann, als mit Monatsangaben, hatte ich zunächst geplant, diese über dem Diagramm anzeigen zu lassen. Nach dem Feedback-Gespräch habe ich mich schließlich dazu entschieden, einen Zeitstrahl links neben das Diagramm zu packen. Dieser zeigt nun den entsprechenden Monat zu den Kalenderwochen an. Dazu verwende ich wieder meine Laufvariable „i“ um zu wissen, wo ich mich im Datensatz gerade befinde. Nach Abzählen der Kalenderwochen im jeweiligen Monat ergibt sich ein Intervall für i, indem dann das Monat gezeichnet werden soll.

Schließlich wollte ich noch die Prozentwerte, die ein Kreissektor einnimmt per Buttonklick anzeigen lassen können. Dies stellt mich vor die Schwierigkeit, dass dieser Text je nach Breite des Sektors an einer anderen Position stehen muss. Diese habe ich mir wie einen Vektor von der Kreismitte bis zum Rand vorgestellt. Zur

```
321 if(zusatzInfos == true){
322   push();
323   textAlign(LEFT);
324   translate(width/2,height/2);
325   let x, y;
326   x=cos(letzterWinkel+radians(anteil)/2)*(durchmesser/2);
327   y=sin(letzterWinkel+radians(anteil)/2)*(durchmesser/2)+20;
328   if(letzterWinkel+radians(anteil)>PI) y=y-30;
329   if(letzterWinkel+radians(anteil)>(PI/2)) { textAlign(RIGHT); x-=20;}
330   if((letzterWinkel+radians(anteil)< 2) && (letzterWinkel+radians(anteil)>1)) y=y+20;
331   let tmp=(ag3*100/gesamt);
332   tmp=runden1(tmp)+"%";
333   text(tmp, x, y);
334   pop();
335 }
```

Berechnung der x, bzw. y Koordinaten verwende ich den Kosinus bzw. Sinus und den entsprechenden Winkel des Sektors. Da der Text in p5 entweder von der linken oder rechten unteren Ecke aus gezeichnet wird, braucht man je nach dem in welchen Kreisviertel sich der Text befindet, eine Korrektur nach unten/oben. Diese werden durch If-Anweisungen auf die Koordinaten addiert. Zur leichteren Ausrichtung des Textes verschiebe ich jedes Mal das Koordinatensystem in die Mitte des Kreises und nach der Textausgabe wieder zurück auf die normalen Startwerte.

#### *Satz von Tufte:*

Der Satz spiegelt sich in der ersten Ansicht des Diagramms wieder. Dabei ist nur das Kuchendiagramm, Legende und der Zeitstrahl zu sehen. Dies reicht aus um die Kernmerkmale der Visualisierung herauszuarbeiten. Es lässt sich ohne weitere Informationen herauslesen, dass sich die Inzidenzen der verschiedenen Altersgruppen über die Zeit hin verändern. Zu Beginn der Pandemie steigt die Inzidenz in der älteren Bevölkerung merklich an. In der Mitte des Jahres 2020 verlagert sich dies in die jüngere Bevölkerung. Gegen Ende des Jahres verschiebt sich das wieder zurück. Zu Beginn des nächsten Jahres ist zu erkennen, dass die Inzidenz der Älteren wieder abnimmt. Dies lässt sich mit der zunehmenden Zahl der Impfungen erklären. All dies lässt sich ohne zusätzliche Information herauslesen.